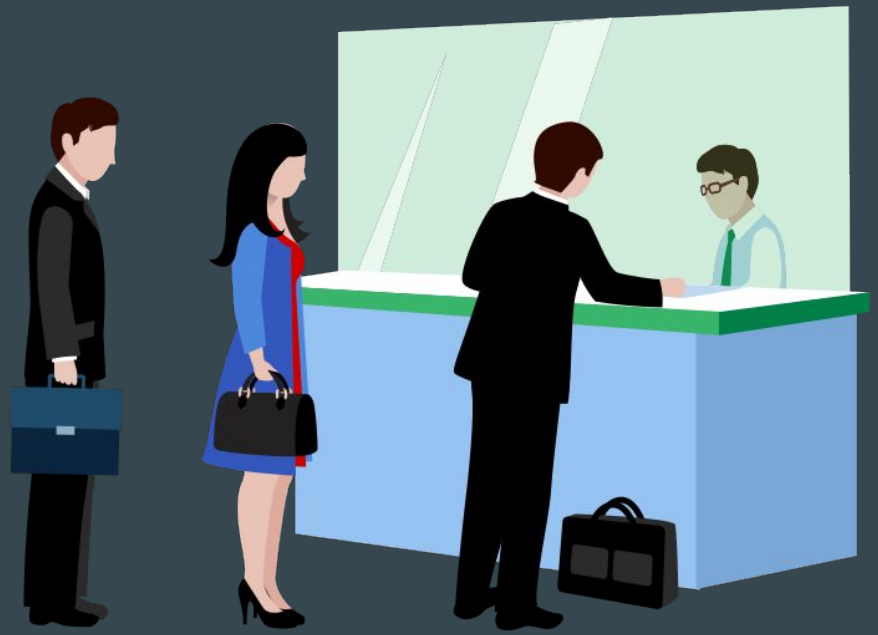
Stacks and queues

Piles et *files* pour les froggies

Queue (file) - FIFO



First In, First Out

When to use: store elements that you want to process later in the **same order**

Queue example

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

How to explore the elements in order of distance from B?

- put B in a queue
- repeat the following:
  - dequeue an element
  - process it
  - enqueue its unvisited neighbors

When the queue is empty, you have seen all elements.

Queue example

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Example of execution:

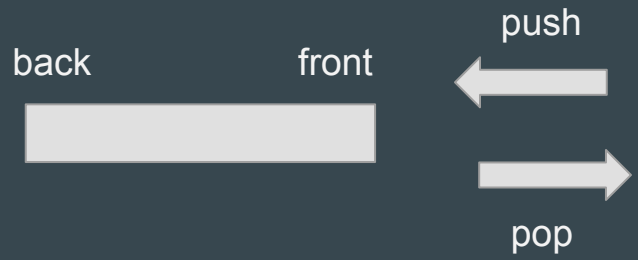
Queue: B
Take B. Queue: AFC
Take A. Queue: FCE
Take F. Queue: CEJ
Take C. Queue: EJGD
Take E. Queue: JGDI
Take J. Queue: GDINK
etc

Stack (pile) - FILO or LIFO

First In, Last Out

When to use: process **most recent items first**, depth-first search, etc

(e.g stacks are used in your browser to come back to previous page)



Stack example

How to know if all the brackets, curly brackets and parentheses in an expression are correctly closed?

→ well... use a stack :)

this one is part of the weekly exercises

{“Python”: (is(), cool[“!”])}

→ correct

{“Java”, (is(), “better”})

→ not correct

Queues in Python

deque in the module `collections` from the standard library

→ optimized to add and remove elements at the front and back in constant time

`append`, `appendleft`, `pop`, `popleft`, `clear`, `rotate`, ...

https://docs.python.org/3/library/collections.html#collections.deque

Stacks in Python

You can use a simple Python **`list`**

→ insertions and deletions at the end are optimized (constant time on average)

`append`, `pop`, `clear`

Going further

https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics)

https://en.wikipedia.org/wiki/Stack_(abstract_data_type)

Priority queue: https://en.wikipedia.org/wiki/Priority_queue (we'll talk about it later)

Credits

- Slides by Louis Sugy for INSAlgo
- Images: Freepik